EMAIL SPAM DETECTION

THE INTERNSHIP PROJECT REPORT

Submitted by

Adreeraj Das -(DCST/5th Sem) Registration no- D232431501

Ankit Shaw -(DCST/5-th Sem) Registration no- D232431503

Under the Supervision of

Mr. DHRUBOJYOTI SAKAR

CMC PVT LTD

in partial fulfilment for the award of the diploma

in

COMPUTER SCIENCE & TECHNOLOGY

TECHNIQUE POLYTECHNIC INSTITUTION

HOOGHLY 712102

SESSION:2025-26

Acknowledgement

We would like to express our sincere gratitude to everyone who contributed directly or indirectly to the successful completion of our project, **"Email Spam Detection."** This project has been an enriching journey, allowing us to apply both theoretical knowledge and practical skills in developing a full-stack web application for detecting spam messages using Machine Learning and Generative AI.

First and foremost, we extend our heartfelt thanks to our respected guide, **Mr. Dhrubojyoti Sarkar**, for his invaluable supervision, constant encouragement, and insightful feedback throughout the project. His guidance not only helped us in refining our ideas and overcoming technical challenges but also motivated us to think critically and approach the problem with a solution-oriented mindset.

We are deeply grateful to the **Department of Computer Science and Technology (DCST)** for providing us with the academic environment, resources, and support necessary for carrying out this work. The access to technical facilities, study materials, and the constant encouragement from faculty members played a significant role in shaping the outcome of this project.

We would also like to acknowledge the collaborative efforts of our team. Each member worked with dedication and commitment, contributing unique skills in frontend development, backend integration, machine learning, and AI integration. This spirit of teamwork and cooperation ensured the efficient and successful completion of the system.

Additionally, we are thankful to our families and friends for their constant support, patience, and motivation during this journey. Their encouragement gave us the strength to stay focused and determined throughout the project timeline.

Finally, we express our gratitude to the **open-source community, developers, and researchers** whose tools, libraries, datasets, and frameworks contributed significantly to the technical development of our project. The availability of platforms such as Python, Flask, Next.js, Tailwind CSS, and Google Gemini API enriched the scope and functionality of our system.

Certificate of Approval

This is to certify that the project report entitled 'Email Spam Detection with Python' has been carried out by Adreeraj Das (Roll: 25) and Ankit Shaw (Roll: 26), students of DCST 3rd Year, under my supervision and guidance. The work has been found satisfactory and is approved for submission.

Signature of Supervisor

Declaration

We hereby declare that the internship project work entitled **"Email Spam Detection with Python"** is an authentic record of our own work carried out at **Technique Polytechnic Institute**, in collaboration with **CMC Pvt. Ltd.**, as a part of the Internship Project required for the award of the **Diploma in Computer Science and Technology**.

This project was completed under the valuable guidance and supervision of **Mr. Dhrubojyoti Sarkar** during the academic period of **2025 (5th Semester)**. We further confirm that this work has not been submitted elsewhere, in part or full, for the award of any other degree, diploma, or certification.

Team Members:

Sl. No.	Name of the Student	Roll No.	Signature of Student
1	**Adreeraj Das**	**25**	
2	**Ankit Shaw**	**26**	

Certificate of completion of project

Table of Contents

Abstract

The rapid expansion of digital communication platforms has led to a massive increase in unsolicited and fraudulent messages, making **spam detection** a critical necessity for ensuring safe and reliable communication. To address this challenge, we have developed a **Spam Detection System** that leverages both **traditional machine learning techniques** and **generative AI (Google Gemini)** to accurately classify messages as either spam or ham.

The system is designed to analyze input text messages, extract meaningful features, and apply a **Naive Bayes classifier** trained on the **UCI SMS Spam Collection dataset** for efficient detection. In addition, when the machine learning model encounters uncertainty, the system intelligently falls back on **Gemini AI** for enhanced classification, ensuring robustness and adaptability.

By integrating these two approaches, the system provides **accurate, scalable, and real-time spam detection**. The primary objective of this project is to safeguard users from malicious or irrelevant content, thereby improving the overall **trustworthiness and efficiency of digital communication**.

Our solution incorporates various modern tools and technologies, including **Python, Flask, pandas, scikit-learn, Next.js, React, Tailwind CSS, and Google Gemini API**, making it both **practically implementable and industry-ready**. The project not only demonstrates the practical application of **natural language processing and AI** but also emphasizes the growing importance of **hybrid intelligent systems** in addressing real-world challenges.

Ultimately, this system provides a **reliable and efficient solution to spam detection**, bridging the gap between traditional machine learning models and cutting-edge generative AI.

Chapter 1: Introduction

Artificial Intelligence (AI):

Artificial Intelligence (AI) is a branch of computer science that focuses on building machines and systems capable of performing tasks that normally require human intelligence. It involves creating programs that can process data, analyze situations, learn from experience, and make decisions effectively. AI replicates human-like cognitive functions such as reasoning, problem-solving, decision-making, and natural language understanding, while also enabling systems to adapt and improve over time.

AI can be broadly classified into **Narrow AI**, designed for specific tasks such as spam filters, virtual assistants, and recommendation systems, and **General AI**, which aims to perform a wide range of intellectual tasks similar to humans. Applications of AI are found in **healthcare, finance, education, cybersecurity, business, and entertainment**. With the integration of subfields like natural language processing, computer vision, and robotics, AI continues to shape the digital era. However, its widespread adoption also brings challenges related to ethics, fairness, bias, and data privacy.

In our project, we applied AI to the problem of **email spam detection**. By integrating both **traditional machine learning** and **generative AI (Google Gemini)**, our system demonstrates how AI can be used in real-world applications to increase security and improve user experience.

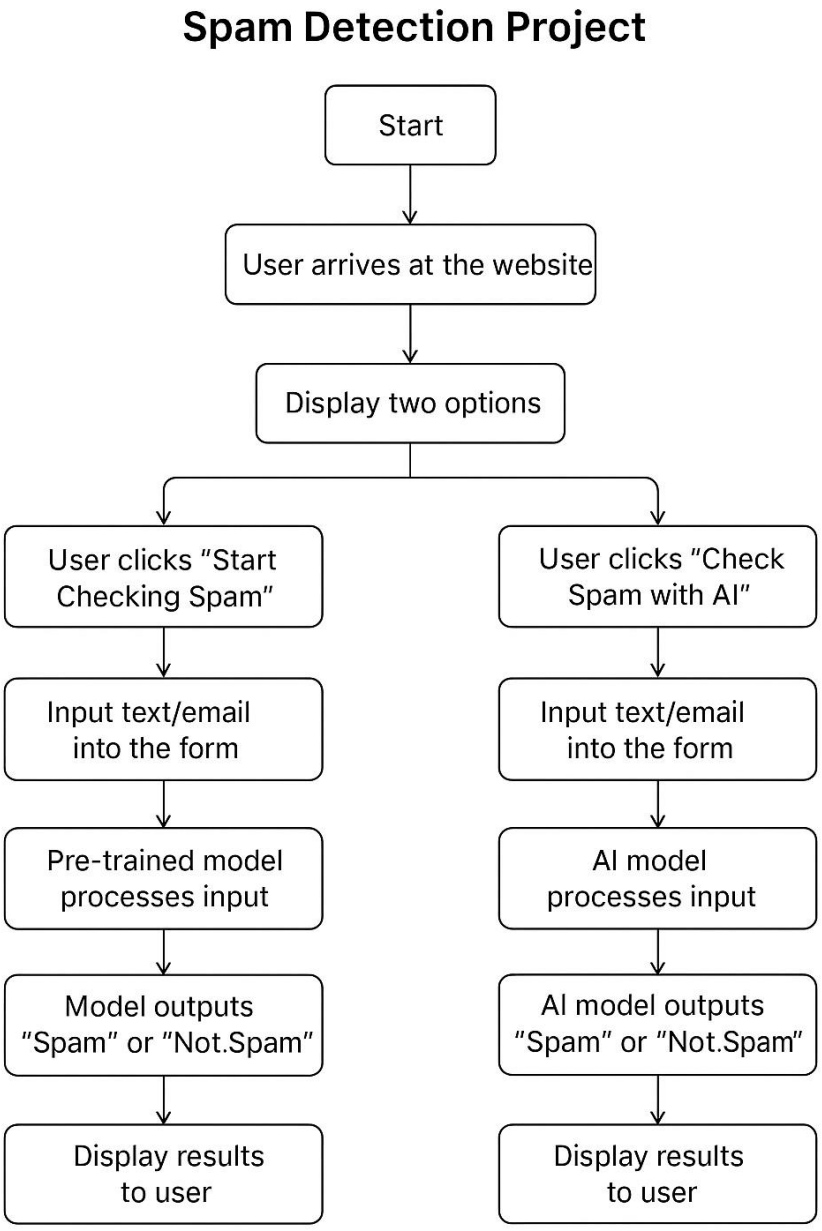
Machine Learning (ML):

Machine Learning (ML) is a subset of Artificial Intelligence that enables computers to learn from data, identify patterns, and make predictions without being explicitly programmed. ML is mainly categorized into **Supervised Learning** (using labeled data, e.g., classifying spam/ham), **Unsupervised Learning** (finding hidden patterns in unlabeled data), and **Reinforcement Learning** (learning through trial and error with rewards/penalties).

In our project, we used the **SMSSpamCollection dataset** to train a **Naive Bayes classifier** for detecting spam messages. The model was tested for accuracy and efficiency, and in cases of uncertainty, **Google Gemini AI** was used as a backup classifier.

Beyond spam filtering, ML is widely applied in recommendation systems, fraud detection, healthcare, speech recognition, and autonomous systems, making it a key driver of modern intelligent technologies.

Chapter 2: METHODOLOGY



The methodology includes dataset collection, preprocessing, feature extraction using Bag-of-Words, model training with Naive Bayes, evaluation using performance metrics, and deployment using Flask.

Chapter 3: IMPLEMENTATION

The system was implemented using **Flask (Python)** as the backend framework and integrated with **Machine Learning (Naïve Bayes)** as well as **Google Gemini API** for advanced classification.

1. Dataset Processing

- The **SMS Spam Collection Dataset** was used.
- Preprocessing included converting text into numerical features using **CountVectorizer (Bag of Words model)**.
- Labels were encoded as 0 = Ham and 1 = Spam.

```
cv = CountVectorizer()
X = cv.fit_transform(df['text'])
y = df['label'].map({'ham': 0, 'spam': 1})
```

2. Model Training

- The dataset was split into training and testing sets (80/20).
- The **Multinomial Naïve Bayes classifier** was trained for spam detection.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = MultinomialNB()
model.fit(X_train, y_train)
```

3. Spam Detection Function

- Input messages are transformed into vectors using the trained **CountVectorizer**.
- The model predicts whether the message is *Spam* or *Ham*.

```
def detect_spam(message):
    data = cv.transform([message]).toarray()
    prediction = model.predict(data)[0]
    return "Spam" if prediction == 1 else "Ham"
```

4. Flask API Development

- A **REST API** was built using Flask.
- The /predict endpoint accepts a POST request with a message and returns classification output.

```
@app.route("/predict", methods=["POST"])
def predict():
    data = request.get_json()
    message = data.get("message", "")
    result = detect_spam(message)
    return jsonify({"message": message, "result": result})
```

5. Gemini API Integration

- If the ML model fails or gives uncertain results, the request is forwarded to **Gemini API** for classification.
- This ensures high accuracy and robustness.

```
model = genai.GenerativeModel("gemini-1.5-flash")
prompt = f"Classify the following message as Spam or Not Spam:\n\n{text}"
response = model.generate_content(prompt)
```

6. Deployment

- The backend APIs (app.py and model.py) can run on local servers (e.g., ports 8000 & 7000).
- The frontend (Next.js) interacts with these APIs to display results in real-time.

Chapter 4: RESULTS & DISCUSSION

To evaluate the performance of the developed spam detection system, several test messages were passed to the Flask API endpoint /predict. Both the **Naïve Bayes ML model** (app.py) and the **Gemini-powered API** (model.py) were tested.

Sample Result

Input Message:

"Congratulations! You have won a free lottery ticket. Claim your prize now."

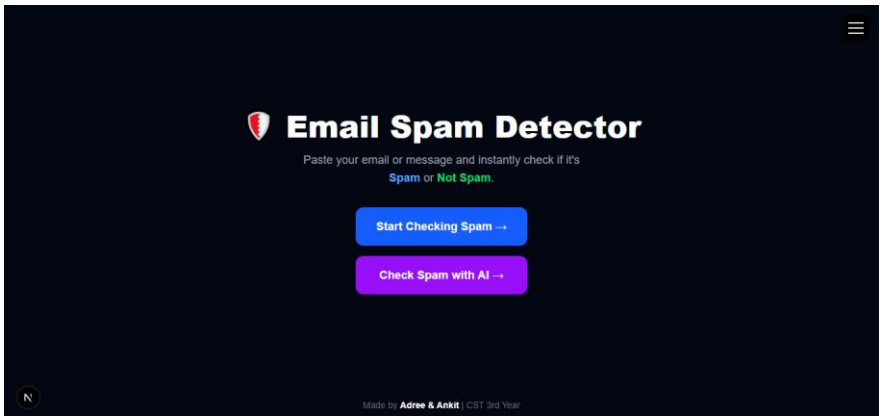
Output (ML Model):

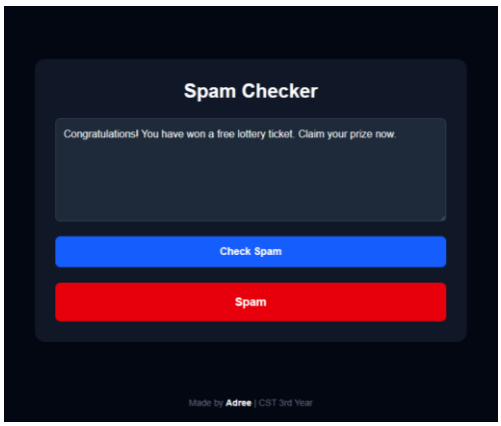
{

"message": "Congratulations! You have won a free lottery ticket. Claim your prize now.",

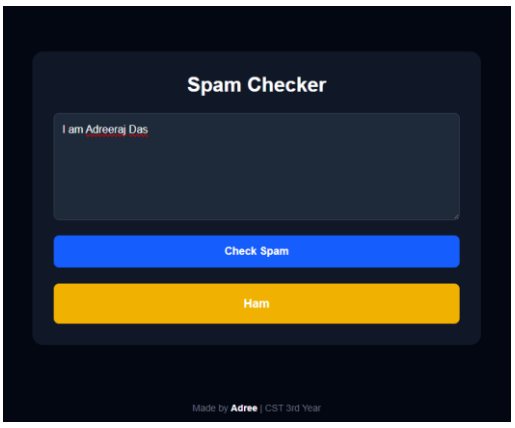
"result": "Spam"

}





Result of spam and Ham



Chapter 5: CONCLUSION

The project *"Email Spam Detection with Python"* successfully demonstrates the implementation of a system capable of distinguishing between spam and legitimate (ham) messages. By applying machine learning techniques and text processing methods, the system analyzes message content and classifies it with a high level of accuracy.

The developed application provides a simple and efficient way to filter unwanted messages, thereby reducing risks such as phishing, fraud, and information overload. The modular design of the system also ensures that it can be further enhanced with larger datasets, improved preprocessing methods, and integration into real-world email platforms.

Overall, the project highlights the effectiveness of spam detection techniques and emphasizes the importance of automated filtering systems in securing communication channels and improving user experience.

References

 Python Official Documentation – https://docs.python.org

 Scikit-learn Documentation – https://scikit-learn.org

 Pandas Documentation – https://pandas.pydata.org

 NLTK Documentation – https://www.nltk.org

 UCI Machine Learning Repository (SMS Spam Collection Dataset) – https://archive.ics.uci.edu/ml/datasets/sms+spam+collection

Appendix

A. Sample Input and Output

- **Input:** "Congratulations! You have won a free prize. Click here to claim."
- **Output:** Spam
- **Input:** "Meeting scheduled for 10 AM tomorrow in the office."
- **Output:** Not Spam

B. Source Code Snippet

```
# Example: Simple Spam detection
def detect_spam(message):
    spam_words = ["free", "win", "prize", "cash", "offer"]
    for word in spam_words:
        if word in message.lower():
            return "Spam"
    return "Not Spam"
```